基于自适应噪声添加的防御对抗样本的算法 *

刘　野，黄贤英†，刘文星，朱小飞，李昭平
(重庆理工大学 计算机科学与工程学院，重庆 400054)

摘　要：近年来，基于深度神经网络的图像分类技术已经取得了巨大的成功，然而，最近研究表明深度神经网络容易受到对抗样本的攻击。为了解决这个问题，一些工作通过向图像中添加高斯噪声来训练网络，从而提高网络防御对抗样本的能力，但是该方法在添加噪声时并没有考虑到神经网络对图像中不同区域的敏感性是不同的。针对这一问题，提出了梯度指导噪声添加的对抗训练算法。该算法在训练网络时，根据图像中不同区域的敏感性向其添加自适应的噪声，在敏感性较大的区域上添加较大的噪声，抑制网络对图像变化的敏感程度，在敏感性较小的区域上添加较小的噪声，提高其分类精度。在 cifar-10 数据集上与现有的算法进行比较，实验结果表明，提出的方法有效地提高了神经网络在分类对抗样本时的准确率。

关键词：深度神经网络；图像分类；对抗样本；自适应噪声
中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2020.03.0055

Algorithm for defense adversarial example based on adaptive noise addition

Liu Ye, Huang Xianying†, Liu Wenxing, Zhu Xiaofei, Li Zhaoping
(*School of Computer Science & Engineering*, *Chongqing University of Technology*, *Chongqing* 400054, *China*)

Abstract: Image classification techniques based on deep neural networks have achieved great success in recent years. However, recent studies have shown that deep neural network are vulnerable to the attack of adversarial examples. To solve this problem, some works train networks by adding Gaussian noise to the image. Thereby improving the ability of the network to defend adversarial examples, but the method does not consider the sensitivity of the network to different areas in the image when adding noise. To solve this problem, this paper proposed an adversarial training algorithm based on gradient guidance noise addition. When training the network, adding adaptive noise to different areas based on the sensitivity, adding large noise to the more sensitive areas, suppressing the sensitivity of the network to image changes, adding less noise to the less sensitive areas and improves the network classification accuracy. Compared with the existing algorithms on the cifar-10 dataset, the experimental results show that the proposed method effectively improved the accuracy of neural networks when classifying adversarial examples.

Key words: deep neural networks; image classification; adversarial example; adaptive noise

0　引言

近年来，深度神经网络(deep neural network，DNN)在各种应用中都取得了巨大的成功，包括图像分类[1]，语音识别[2]，机器翻译[3]，自动驾驶[4]，图像字幕[5]以及对象识别[6]。然而，2014 年 Christian Szegedy 等人发现深度神经网络具有很容易受到对抗样本攻击的特性，在图像分类任务中，给定一个正确分类的图像，向图像添加精心设计的微小扰动可以使深度神经网络以较高的置信度分类错误，这样的添加有扰动的图像被称为对抗样本[7]。除了图像分类以外，其他 DNN 的应用也受到了对抗样本的攻击，如视觉问题问答[8, 9]，图像字幕[10]，语义分割[11]及其他[26, 27]等，这对深度神经网络的应用构成了一定的威胁。

图像分类技术是深度神经网络在计算机视觉的各种应用中的基础任务，具有极其广泛的应用，但也是遭受对抗样本攻击较严重的领域。为了提高神经网络正确分类对抗样本(也就是防御对抗样本)的能力，最近，一些工作[12, 13]从模型正则化的角度来考虑，其主要思想是：在训练阶段，通过向输入图像中添加高斯噪声训练神经网络，实现网络正则化，从而提高网络分类对抗样本的准确率。然而这些方法训练时向图像中添加的噪声是从同一高斯分布(相同的均值和标准差)中采样的，并没有考虑到网络对图像中不同像素的敏感性是不同的[14]，即改变输入图像中不同像素，对分类结果的影响大小是不同的。另一些工作提出了对抗训练的方法[15, 16]，该方法的基本思想是：通过在训练过程中，使用投影梯度下降方法[15]生成对抗样本，并将生成的对抗样本加入到训练集中训练网络，从而提高网络防御对抗样本的能力。但是该方法由于标签泄露的问题[17]，为了使训练的网络具有良好的防御对抗样本的能力，在生成对抗样本时需要多次计算输入图像的梯度，导致训练的时间花销达到正常训练的 10 倍以上。

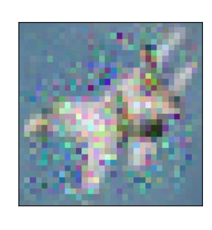
本文将对抗训练思想以及利用噪声图像来训练网络的方法相结合，提出了梯度指导噪声添加(gradient guide noise addition，GGNA)的对抗训练算法。该算法的基本思想是：由于对抗样本改变了原始图像的像素，为了降低网络对输入图像的像素变化的敏感性，在使用噪声图像训练网络时，应在敏感性(梯度值)较大的像素上添加更多的噪声，更有效地降低网络对该像素变化的敏感性，从而提高网络防御对抗样本的能力。而在敏感性较小的像素上添加较小的噪声，提高网

收稿日期：2020-03-30；**修回日期：**2020-05-18　**基金项目：**国家自然科学基金资助项目(17XXW005)；重庆市基础科学与前沿技术研究重点专项资助项目(cstc2017jcyjBX0059)；重庆市巴南区科技计划资助项目(2018TJ05)

作者简介：刘野(1994-)，男，四川宜宾人，硕士研究生，主要研究方向为深度学习，计算机视觉；黄贤英(1967-)，女(通信作者)，重庆人，教授，硕导，硕士，主要研究方向为计算机应用(wldsj_cqut@163.com)；刘文星(1995-)，男，重庆人，硕士研究生，主要研究方向为深度学习；朱小飞(1979-)，男，重庆人，教授，硕导，博士，主要研究方向为深度学习，自然语言处理；李昭平(1995-)，男，四川人，硕士研究生，主要研究方向为深度学习.

络分类精度。具体的说，在训练阶段，本算法首先计算输入图像中各像素的梯度值，再用归一化方法将梯度大小转换为高斯分布的标准差(高斯分布的均值固定为 0)，输入图像中不同像素添加的噪声将从对应的标准差的高斯分布中独立采样。并将该添加有噪声的图像加入训练集中训练网络。在测试对抗样本时，也向对抗样本中加入一定的噪声再测试对抗样本。


(a) 原图


(b) 自适应噪声图

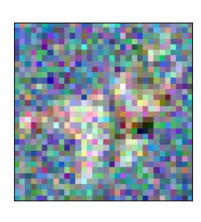

(c) 普通噪声图

图 1　原图与对应的噪声图

Fig. 1　Image and corresponding noise image

图 1 显示了一张图像与该图像的自适应噪声图与普通噪声图的不同。其中，(a)为 cifar-10 数据集中的一张图像，(b)为添加有自适应噪声的图，该噪声是根据模型对图像中不同像素的敏感性而添加，(c)为添加普通的高斯噪声图。

本文的主要贡献点如下：

a) 为提高模型正确分类对抗样本的能力，将使用噪声图像训练的网络方法和对抗训练方法相结合，提出了一种新的防御对抗样本的算法；

b) 提出的 GGNA 方法在用添加噪声的图像训练网络的过程中，考虑了网络对不同像素的敏感性，实现了自适应的噪声添加；

c) 提出的 GGNA 方法使用了对抗训练的思想，但在训练过程中只需计算一次图像的梯度，相对于普通的对抗训练方法，减少了训练的时间。

1　相关工作

1.1　对抗攻击

最近，在对抗样本生成(也就是对抗攻击)方面有了许多的研究成果。通常，根据暴露给攻击者神经网络的信息的多少，对抗攻击可以分为白盒攻击(PGD[15]，FGSM[7] 和 Deepfool[18])和黑盒攻击。对于白盒攻击，攻击者能获得有关神经网络的所有信息，包括网络结构和网络权重，可以通过反向传播计算输入图像的梯度，梯度对于攻击者非常有用，因为梯度代表了输出对于输入图像的敏感性，攻击者根据梯度方向修改图像上的像素，从而生成对抗样本。而对于黑盒攻击[19]，攻击者只知道网络的外部信息(如输入和输出)，通过对抗样本的转移性进行攻击。由于白盒攻击的信息更丰富，因此，白盒攻击可以得到更高的攻击成功率[20]。常见的攻击方法有如下几种：

快速梯度符号方法[7](fast gradient sign method，FGSM)是一种有效的单步对抗攻击方法。其基本思想是：给定一个输入向量和相应的攻击目标 t，FGSM 沿着测试 loss 关于向量 X 的每个元素的梯度方向改变每个元素 x，对抗样本的生成可描述如下：

$$\hat{x}=x+\varepsilon \bullet sign(\nabla_x L(f_\theta(x),t)) \tag{1}$$

其中 ε 是决定攻击强度的总的约束扰动。$f_\theta(x)$ 是计算输入 x 时参数为 θ 的 DNN 时的输出，$sign(\bullet)$ 是符号函数。注意，若原始样本 $x\in[0,1]$，则攻击所生成的对抗样本要做一个剪切以确保 $\hat{x}\in[0,1]$。

投影梯度下降方法(projected gradient descent，PGD)[15]是 FGSM 攻击方法的多步变体。其基本思想是：用 $\hat{x}^{k-1}=x$ 作为初始化，迭代多次计算输入 $\hat{x}$ 的梯度值，更新对抗样本，迭代过程可以描述如下：

$$\hat{x}^k=\prod P_\varepsilon(x)(\hat{x}^{k-1}+\alpha \bullet sign(\nabla_x L(f_\theta(\hat{x}^{k-1}),t))) \tag{2}$$

其中 $P_\varepsilon(x)$ 是以 $x\pm\varepsilon$ 为上下界的投影空间，ε 是总的约束扰动的大小，α 是每迭代一次扰动的步长大小，k 为迭代的次数。在白盒对抗攻击中 PGD 是一个非常强大的攻击方法，相对于 FGSM 方法，PGD 攻击产生的对抗样本更容易使模型错误分类。

Deepfool 攻击方法[18]是 Moosavi-Dezfooli 等人为改进了 FGSM 中扰动大小需要人为选择的问题而提出的对抗样本生成算法，该方法可以通过多次迭代求解以下的优化问题，直到得出满足 $f_\theta(\hat{x})\neq f_\theta(x)$ 条件的扰动大小：

$$\hat{x}^{k+1}=\hat{x}^k-\frac{f_\theta(\hat{x}^k)}{\|\nabla_x L(f_\theta(\hat{x}^k),t)\|_2^2}\nabla_x L(f_\theta(\hat{x}^k),t) \tag{3}$$

其中 x 是正常图像，$\hat{x}$ 是产生的对抗样本，$f_\theta(x)$ 是分类器，Deepfool 攻击是对决策边界进行攻击，相对于 FGSM 攻击，Deepfool 攻击所产生的扰动更小。

1.2　对抗防御

针对对抗样本的攻击，最近在图像分类的防御对抗样本(对抗防御)方面也提出了很多方法，包括对抗样本检测方法，对抗训练方法和基于正则化的方法等。Pang 等[22]提出了一种对抗样本检测方法，该方法通过最小化反向交叉熵，鼓励神经网络学习图像的潜在表示，从而将对抗样本和正常样本分开。虽然对抗样本检测算法易于实现且检测成功率高，但是该方法只检测输入图像是否为对抗样本，对于对抗样本的正确分类仍然需要和其他的防御方法共同使用才能实现。Marday 等[15]通过对抗训练的方法来提高网络防御对抗样本的能力，对抗训练的过程在理论上相当于求解如下的最大最小值问题：

$$\min_f \mathrm{E}\left\{\max_{\hat{x}\in \mathrm{B}(x,\varepsilon)} L(f_\theta(\hat{x}),Y)\right\} \tag{4}$$

其中，Y 是对应的标签，$L(\bullet)$ 代表 loss 函数，x 是原始样本，$\hat{x}$ 是对抗样本，$B(x,\varepsilon)$ 是以 $x\pm\varepsilon$ 作为上下界的空间，$f_\theta(\bullet)$ 是神经网络分类器，θ 是分类器的参数，在求解 $\max L(f_\theta(\hat{x}),Y)$ 的内部最大化问题时是通过 PGD 攻击方法生成对抗样本来近似求解，而 $\min E\{\bullet\}$ 的外部最小化问题，则通过更新网络模型的参数以最小化内部对抗样本所引起的对抗性损失值。最新的 TRADES[16]方法改进了对抗训练的方式，将对抗训练过程看做近似求解如下的最大最小值问题：

$$\min_f E\left\{L(f_\theta(x),Y)+\alpha \bullet \max_{\hat{x}\in B(x,\varepsilon)} L(f_\theta(\hat{x}),f_\theta(x))\right\} \tag{5}$$

其中 α 为正则化项，该方法权衡了正常样本和对抗样本的准确率并得到了更好的防御对抗样本的效果，在求 $\max L(f_\theta(\hat{x}),f_\theta(x))$ 的内部最大化的问题时，TRADES 方法是也是通过 PGD 方法多次迭代生成对抗样本近似求解。Zantedeschi 等[12]通过在输入图像中添加高斯噪声进行数据增强来训练网络实现网络正则化，从而降低神经网络对输入变化的敏感性，在测试对抗样本时也向对抗样本中添加噪声。但是该方法在训练时加入的噪声为普通的高斯噪声，没有考虑梯度信息。Liu 等[13]为防御对抗样本提出向输入及网络中添加高斯噪声训练网络的方法。Wang 等[25]提出了 MART 方法,该方法通过区分训练过程中分类错误和正确的样本，并对分类错误和正确的样本采用不同的最大化方法来训练鲁棒的分类模型。

2　梯度指导噪声添加(GGNA)对抗训练算法

2.1　梯度与添加噪声的关系分析

在图像分类的神经网络中，输出对于输入图像中的每一像素的梯度值不同，则每一像素对输出的影响大小也不同。

梯度值较大的像素，则该像素对输出的影响也较大，即使对这些像素做很小的改变也很容易使原本分类正确的图像分类错误，而对于梯度值接近 0 的像素，对该像素做较大的扰动对分类结果也几乎无影响。图 2 给出了对于某一神经网络分类器 f，输入图像的某一个通道 8×8 区域上梯度绝对值大小，可以看出不同像素的梯度相差较大。

对抗样本是对原始图像的某些像素做了改变的图像，为了更有效地降低输入图像中像素变化对输出的影响，在使用噪声图像训练模型时，应在梯度较大的像素上添加更多的噪声，抑制神经网络对该像素的敏感性，而在梯度接近 0 的像素上添加更少的噪声，提高网络分类精度。由于添加的噪声是从高斯分布中独立采样，在高斯分布的均值设为 0 时，标准差越大则采样到的更大的噪声的概率也越大。基于以上的分析，在使用噪声图像训练模型时，可以将输入图像的梯度转换为高斯分布的标准差，而各像素中添加的噪声将从该像素的梯度转换的标准差的高斯分布中独立采样，梯度较大的像素对应的噪声从标准差较大的高斯分布中采样，梯度较小的像素对应的噪声从标准差较小的高斯分布中采样。

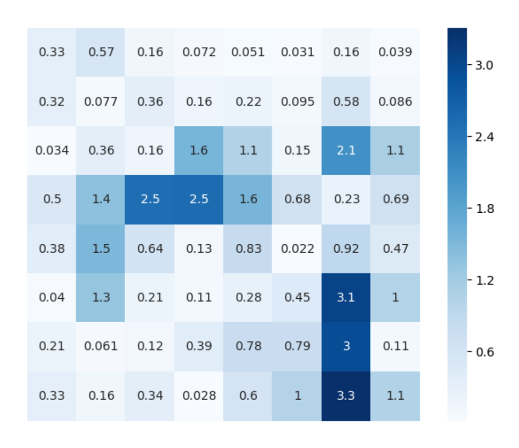


图 2　输入图像某一通道的梯度绝对值

Fig. 2　Gradient absolute value of the image in a channel

2.2　算法理论分析

在基于神经网络的图像分类任务中，训练一个普通的分类模型，为了获得高的分类准确率，需要最小化模型的损失函数：

$$\min_f E\left\{L\left(f_\theta(x),Y\right)\right\} \tag{6}$$

然而，由于对抗样本的存在，为了让模型不仅对原始样本具有较高的分类准确率，对于对抗样本也要有较高的分类准确率，最小化损失函数：

$$\min_f E\left\{L\left(f_\theta(x),Y\right)+\max_{\hat{x}\in B(x,\varepsilon)} L\left(f_\theta(\hat{x}),Y\right)\right\} \tag{7}$$

其中 $\max_{\hat{x}\in B(x,\varepsilon)} L(f_\theta(\hat{x}),Y)$ 是通过在训练过程中生成使得模型损失值最大的对抗样本 $\hat{x}$ 来近似实现，常见实现方式如下：

$$\hat{x}=x+N(0,\eta^2) \tag{8}$$

$$\hat{x}^k=\hat{x}^{k-1}+\alpha\bullet sign(\nabla_x L(f_\theta(\hat{x}^{k-1}),Y)) \tag{9}$$

其中，$N(0,\eta^2)$ 表示从均值为 0，标准差为 η 的高斯分布中采样，即式(8)表示向图像中加入同分布的高斯噪声。而式(9)则表示沿着梯度方向多次迭代修改图像以生成对抗样本，其中 k 表示迭代的次数，α 表示每一次修改的大小，$sign(\bullet)$ 表示符号函数，$\nabla_x L(f_\theta(\hat{x}^{k-1}),Y)$ 表示输入图像的梯度。结合以上两个公式并根据 2.1 节中关于梯度和添加噪声的关系分析，得出以下的对抗样本生成方法：

$$\hat{x}\leftarrow x+N\left(0,\left|\nabla_x L(f_\theta(x),Y)\right|^2\right) \tag{10}$$

该方法相对于式(8)中的方法，增加了梯度的信息，使添加的噪声和梯度相结合。相对于式(9)中的方法，在获得梯度信息的同时使得迭代的次数从 k 次降低到了 1 次，而式(9)中的方法需要迭代 k 次。受文献[16]的启发，算法整体的训练过程如下：

$$\min_f E\left\{L(f_\theta(x),Y)+L\left(f_\theta\left(x+N\left(0,|g|^2\right)\right),f_\theta(x)\right)\right\} \tag{11}$$

其中，$g=\nabla_x L(f_\theta(x+N(0,\alpha^2)),f_\theta(x))$。

2.2.1 算法时间复杂性分析

由式(8)(9)和(10)可知，由于式(8)的方法不需要计算梯度值，且只需要添加普通高斯噪声，所以花费的训练时间最少，而式(9)中的方法需要迭代 k 次反向传播以计算输入图像的梯度值，在训练深度学习模型中由于参数量很大，反向传播需要花较多的时间，所以该方法所花费的训练时间较多。而提出的式(10)中的方法虽然同样需要计算梯度值，但是只需要计算一次，减少了多次反向传播计算梯度值的时间，并将梯度转换为高斯分布的标准差，达到了添加自适应噪声的效果。总体算法的时间复杂度为式(8)<式(10)<式(9)。

2.3　算法描述

本小节阐述了本文提出的梯度指导噪声添加的对抗训练算法的具体步骤。在训练网络阶段，首先计算原始输入图像 x 中各像素的梯度 g，再将该梯度转换为高斯分布的标准差 σ，各像素添加的噪声将从对应标准差的高斯分布中独立采样，并将该添加有噪声图像加入训练集中训练网络，直到网络收敛。在测试对抗样本时也向对抗样本中添加噪声，并进行多次预测，投票得出最终结果。详细步骤如下:

2.3.1 训练网络阶段

a) 计算图像 x 中各像素的梯度 g：算法利用对抗训练的思想，根据式(5)，计算输入图像 x 的梯度 $g=\nabla_{\hat{x}} L(f_\theta(x),f_\theta(\hat{x}))$，其中 $L(f_\theta(x),f_\theta(\hat{x}))$ 代表输入图像 x 与初始噪声图像 $\hat{x}$ 输出之间的相对熵，为了计算该相对熵，将噪声图像初始化为 $\hat{x}=x+0.001\bullet N(0,1)$，其中 $N(0,1)$ 为均值为 0，标准差为 1 的高斯分布。

b) 将梯度 g 转换为高斯分布的标准差 σ：由于梯度值较大则对应的高斯分布的标准差 σ 也应该较大，且标准差 $\sigma\geq 0$，所以先对梯度取绝对值即 $g\leftarrow|g|$，再对其做最大最小归一化，即 $g\leftarrow(g-\min(g))/(\max(g)-\min(g)+0.0001)$，其中为了防止除 0 操作，在分母中加入了 0.0001 的权重，并归一化到[0，1]。但归一化后各像素梯度大小的差距仍然会达到 1000 倍以上，为了使得梯度转换得到的标准差的大小更加稳定，将归一化以后的梯度再除以梯度的均值，并裁剪到[0-1]范围中，为了控制高斯分布的最大标准差，还需乘以 $\alpha\in[0,1]$ 超参数，即 $g\leftarrow clamp(g/(mean(g)+0.0001),0,1)\times\alpha$,其中 $clamp(\bullet,0,1)$ 为裁剪到[0, 1]中,最后将计算得到的梯度 g 作为高斯分布的标准差 σ。

c) 添加自适应噪声得到噪声图像 $\hat{x}$：由于输入图像 x 中不同像素的梯度不同，则各像素添加的噪声也从不同的高斯分布(标准差不同)中独立采样，即 $\hat{x}\leftarrow clamp(\hat{x}+N(0,\sigma^2),0,1)$，其中 $\sigma\in R^{h\times w\times 3}$。

d) 计算 loss 值更新网络参数 θ：根据式(5)计算 loss 值，该 loss 值由两部分组成，第一个是输出 $f_\theta(x)$ 和标签 Y 之间的交叉熵 $L(f_\theta(\hat{x}),Y)$，为了提高正常样本的准确率。第二个 $L(f_\theta(x),f_\theta(\hat{x}))$ 为噪声图像 $\hat{x}$ 和正常图像 x 之间的相对熵，通过最小化相对熵的值，使网络能把噪声样本和正常样本分类为一样，即正常样本正确也能分类正确。然后更新参数 $\theta\leftarrow\theta-\beta\bullet\nabla_\theta[L(f_\theta(x),Y)+L(f_\theta(x),f_\theta(\hat{x}))]$，其中 θ 为网络参数，β 为学习率，直到网络收敛。

2.3.2 测试对抗样本阶段

该算法在测试对抗样本时，也向测试的对抗样本中添加噪声，但由于测试时是没有梯度信息的，所以只向测试的对抗样本中添加普通高斯噪声。在知道对抗样本的扰动大小时，

高斯噪声的大小可以根据对抗样本的扰动大小进行相应调整，为简单起见，测试对抗样本时添加的噪声的标准差也可以设置为训练时添加的实际噪声的标准差的均值。并集成多次测试结果，投票得出最终预测结果，在测试正常样本时不添加噪声。

梯度指导噪声添加(GGNA)的对抗训练方法的伪代码如算法 1 所述。

算法 1　梯度指导噪声添加的对抗训练算法

输入：高斯噪声标准差 σ_1，学习率 β，批量大小 m，网络参数 θ。

输出：训练完成的网络 f_θ。

a) 从数据集中读取 m 个样本 $s=\{x_1,...,x_m\}$

b) for $i=1,...,m$ do

c) $\hat{x}_i \leftarrow x_i + 0.001 \cdot N(0,1)$

d) $g_i \leftarrow abs(\nabla_{\hat{x}_i} L(f_\theta(x_i), f_\theta(\hat{x}_i)))$

/* 其中 $abs(\bullet)$ 为取绝对值，$\nabla_{\hat{x}_i} L(\bullet)$ 为计算输入图像的梯度。 */

e) $g_i \leftarrow (g_i - \min(g_i))/(\max(g_i) - \min(g_i) + 0.0001)$　/* $\min(\bullet)$ 为取最小值，$\max(\bullet)$ 为取最大值。 */

f) $g_i \leftarrow clamp(g_i/(mean(g_i)+0.0001),0,1) * \sigma_1$

/* 其中，$clamp(\bullet)$ 为裁剪到 $[0,1]$ 中，$mean(\bullet)$ 为取均值。 */

g) $\hat{x}_i \leftarrow clamp(\hat{x}_i + N(0,g_i^2),0,1)$　// 添加噪声，并裁剪到 $[0,1]$。

h) end for

i) $\theta \leftarrow \theta - \beta \bullet \sum_{i=1}^{m} \nabla_\theta [L(f_\theta(x_i), y_i) + L(f_\theta(x_i), f_\theta(\hat{x}_i))]/m$

/* 利用计算得到的 loss 值更新参数。 */

j) 重复执行步骤 a)至步骤 i)，直到网络收敛

3　实验

3.1　实验设置

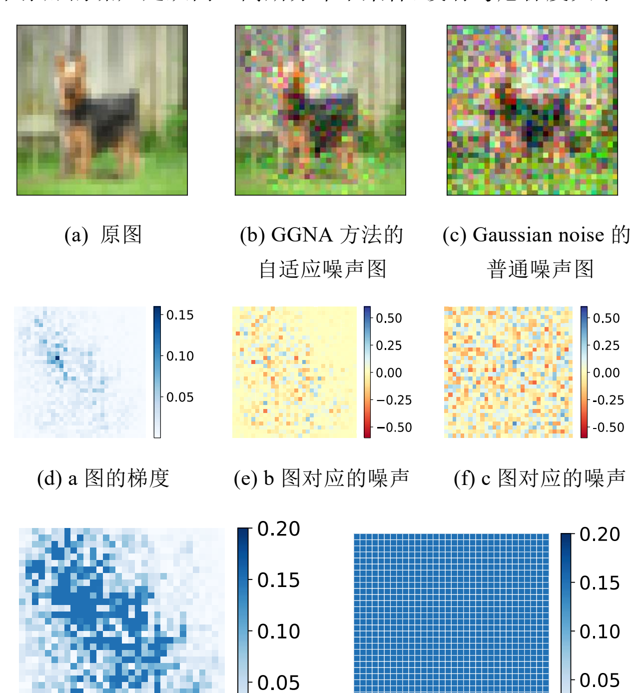
实验使用公开的 cifar-10 数据集[23]。该数据集中，有 5 万张训练样本图像和 1 万张测试样本图像，每张图片是 32×32 大小的彩色图片。训练时使用的数据增强方式为：在图像的上下左右分别填充 4 个像素点 0，然后再随机裁剪为 32×32 大小的图像，以及概率为 0.5 的随机水平翻转，并将数据除以 255 归一化到[0, 1]。实验采用经典的 resnet-18 网络结构[14]。

训练过程中使用的每个批次大小为 64，优化器是 SGD 优化器，初始的学习率为 0.1，在第 30 周期的时候将学习率降为 0.01，在第 40 周期的时候将学习率降为 0.001，一共训练 50 个周期。对于 TRADES 方法，在训练时使用的扰动大小为 0.031(8/255)，即每个像素点的值最大改变为 8，迭代次数为 10 次，扰动步长为 0.0031，标记为 TRADES(10)。对于 GGNA 方法，在训练时加入的噪声的最大标准差初始化为 0.1，在第 10，20 和 25 周期的时候分别将标准差提升到 0.15，0.20 和 0.25，在第 30 周期时将标准差修改为[0.25, 0.35]的范围，分别训练多个模型对比测试。对于以前的利用噪声图像训练网络的方法，标记为 Gaussian noise，该方法在训练时向输入图像中添加普通高斯噪声训练模型。对于 MART 方法，使用和 TRADES 方法相同的参数配置。

实验使用 FGSM，PGD，Deepfool 攻击方法进行广泛的测试，在测试对抗样本时为减少随机性，将实验的测试次数设置为 50 次。实验使用的 CPU 是 i7-9700K，GPU 是 NVIDIA RTX 2080Ti。

3.2　GGNA 方法与之前方法在添加噪声的不同

图 3 显示了高斯噪声的最大标准差为 0.15 时，在训练阶段，梯度指导噪声添加(GGNA)的对抗训练方法与之前使用噪声图像训练方法在添加噪声时的不同，其中(a)是 cifar-10 中训练的原图，(b)是 GGNA 方法训练时产生的自适应噪声图。(c)是普通的添加高斯噪声的方法训练时产生的噪声图，(d)是 a 图的一个通道的梯度图，(e)是 GGNA 方法向原图中添加的实际噪声，(f)是以前的方法向原图中添加的实际噪声，(g)是 GGNA 产生的噪声标准差，(h)是普通的添加高斯噪声的方法产生的噪声标准差。从该图中可以看出，为了降低训练网络在对输入变化的敏感性，GGNA 方法在训练时实现了自适应的噪声添加，在梯度较大的像素添加了较大的噪声，在梯度较小的像素上添加的噪声也较小。而以前的方法在图中添加的噪声是从同一高斯分布中采样，没有考虑梯度大小。



(a) 原图　(b) GGNA 方法的自适应噪声图　(c) Gaussian noise 的普通噪声图

(d) a 图的梯度　(e) b 图对应的噪声　(f) c 图对应的噪声

(g) b 图对应噪声标准差　(h) c 图对应噪声标准差

图 3　GGNA 方法和之前方法在添加噪声时的不同

Fig. 3　The difference between GGNA and previous method when adding noise

3.3　GGNA 方法训练时添加噪声标准差对准确率的影响

表 1 给出了 GGNA 方法将训练时添加的噪声最大标准差设置为[0.25, 0.35]的范围时，实际添加的噪声的标准差，训练出的模型在测试正常样本时的准确率以及测试对抗样本时的准确率。其中，对抗样本是使用扰动大小 ε 为 8/255，迭代次数 k 为 8，步长 α 为 1/255 的 PGD 攻击产生，为简单起见，在测试对抗样本时添加的噪声标准差为训练阶段实际添加的噪声的标准差的均值，在测试正常样本时不向图像中添加高斯噪声。从表 1 中可以看出，在一定范围内随着训练时添加噪声的标准差的增加，训练出的网络在测试对抗样本时的准确率也在逐渐提高，但在测试正常样本时准确率却在下降，为了权衡两个准确率，在下面的实验中，GGNA 方法采用 0.30 的最大标准差，对应的添加的实际噪声的标准差为 0.18。

表 1　训练时添加的不同标准差的噪声对准确率的影响

Tab. 1　The effect of adding noise with different standard deviations during training on accuracy

最大标准差差	实际标准差差	对抗样本准确率率	正常样本准确率率
0.25	0.15	0.4440	0.8878
0.27	0.16	0.4554	0.8807
0.28	0.17	0.4652	0.8796
0.29	0.17	0.4775	0.8763
0.30	0.18	0.4870	0.8740
0.31	0.18	0.4880	0.8732
0.32	0.19	0.4914	0.8706
0.33	0.19	0.5003	0.8650
0.35	0.20	0.5042	0.8580

3.4　GGNA 方法和普通对抗训练方法训练的时间对比

表 2 显示了 GGNA 与 TRADES(10)在 cifar-10 数据集上训练一个周期的时间对比，实验的参数如 4.1 实验设置所述，TRADES(10)表示在对抗训练时迭代计算 10 次输入图像的梯度以生成对抗样本。由于 GGNA 方法在对抗训练过程中只需要计算 1 次输入图像的梯度，而 TRADES(10)却要迭代多次，相对于 TRADES(10)，GGNA 方法的训练时间只有其 30%，有效地减少了训练的时间。

表 2　GGNA 方法和 TRADES 方法训练时间对比

Tab. 2　Training time between GGNA and TRADES method

方法	一个周期的训练时间
GGNA	118 S
TRADES(10)	385 S

3.5　不同防御方法在正常样本和对抗样本上的准确率比较

由于不同防御方法在对抗样本和正常样本的准确率之间都会存在一定的权衡，即同一防御方法在提高对抗样本准确率时相应的会降低正常样本的准确率[24]。为比较不同防御方法在相同条件下正常样本和对抗样本的准确率，本实验采用了扰动大小 ε 为 8/255，迭代次数 k 为 8，步长大小 α 为 3/255 的 PGD 攻击方法，比较了 Normal、MART、Gaussian noise、TRADES(10)和本文提出的 GGNA 防御方法。其中 GGNA 方法在加入噪声的最大标准差为 0.30 时，实际的生成的噪声的标准差为 0.18，为了公平比较，将之前的噪声添加方法的中添加噪声的标准差设定为 0.18，并标记为 Gaussian noise，对于正常训练的方法，标记为 Normal，其他参数设定如 4.1 实验设置所述。实验中在测试正常样本时，GGNA 方法和 Gaussian noise 方法都不添加高斯噪声，在测试对抗样本时添加标准差为 0.18 的高斯噪声。实验结果如表 3 所示，从表 3 中可以看出 GGNA 方法实现了更好的对抗样本以及正常样本的准确率。

表 3　不同方法在正常样本和对抗样本上的准确率

Tab. 3　The accuracy of different methods on normal and adversarial examples

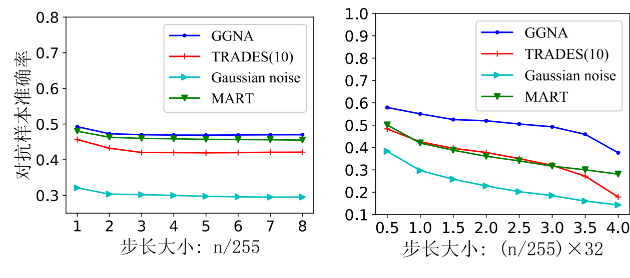
防御方法	对抗样本准确率	正常样本准确率
Normal	0.0000	0.9295
Gaussian noise	0.3030	0.8077
TRADES(10)	0.4104	0.8722
MART	0.4691	0.7622
GGNA	0.4702	0.8740

3.6　不同攻击方法下对抗样本准确率的比较

3.6.1 PGD 攻击

为了更全面的比较 GGNA 方法和其他防御方法在对抗样本准确率上的性能，实验在 PGD 攻击下进行了广泛的测试，包括攻击步长 α，迭代次数 k，最大扰动 ε。根据对抗样本的扰动的计算方法，对抗攻击类型可以分为 L_2 攻击类型以及 L_∞ 攻击类型，在 L_∞ 攻击类型中扰动大小的计算方法为对抗样本 $\hat{x}$ 与原始样本 x 的差的绝对值的最大值，即 $\|\hat{x}-x\|_\infty=\max|\hat{x}-x|$，在 L_2 攻击类型中扰动大小的计算方法为对抗样本 $\hat{x}$ 与原始样本 x 的差的绝对值的平方和再开方，即 $\|\hat{x}-x\|_2=\sqrt{\sum_{i=1}^{N}(\hat{x}-x)^2}$。其中，$N$ 是输入样本每个通道的像素点的个数，在本次实验中 N 为 32×32，对于最大扰动 ε_∞ 为 8/255 的 L_∞ 攻击类型，对应的 L_2 攻击类型的最大扰动 ε_2 为(8/255)×32。在没有特别说明的情况下，本节实验将 L_∞ 攻击类型的最大扰动 ε_∞ 设置为 8/255，将 L_2 攻击类型的最大扰动 ε_2 设置为(4/255)×32，对于 GGNA 方法训练时添加的噪声最大标准差为 0.3，测试时向图像中添加标准差为 0.18 的高斯噪声，对于 Gaussian noise 方法在训练以及测试时都向图像中添加标准差为 0.18 的高斯噪声。实验为了方便比较，将横坐标刻度值设置为 n。其他参数如 4.1 实验设置所述。实验结果如下：

a) 攻击步长 α 对对抗样本准确率的影响：图 4 显示了在迭代次数 k 为 8，且随着步长 α 的增加的 PGD 攻击下，四种防御方法的对抗样本准确率。其中(a)和(b)分别是在 L_∞ 以及 L_2 类型的 PGD 攻击下的准确率，从图 4 中可以看出，随着扰动步长的增加，四种防御方法的对抗样本准确率都有不同程度的下降，但 GGNA 方法相对于其他三种防御方法在 L_∞ 以及 L_2 类型的扰动下都具有更高的对抗样本准确率。

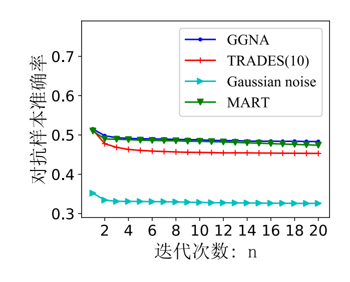


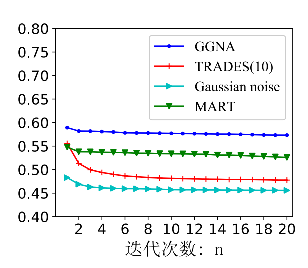
(a)　L_∞ 类型的 PGD 攻击　　(b)　L_2 类型的 PGD 攻击

图 4　随着步长的增加，四种防御方法的对抗样本准确率

Fig. 4　As the step size increases, the accuracy of three defense methods in adversarial examples

b) 攻击迭代次数 k 对对抗样本准确率的影响：图 5 显示了随着 PGD 攻击的迭代次数 k 的增加，四种防御方法的对抗样本准确率的比较，实验将步长 α 设置为 ε/k。其中(a)和(b)分别显示了在 L_∞ 以及 L_2 类型的 PGD 攻击下，四种防御方法的对抗样本准确率变化。可以看出随着迭代次数的增加，在 L_∞ 类型的扰动下，GGNA 方法与 MART 方法取得了相当的对抗样本准确率，相比于另外两种方法取得了更好的对抗样本准确率。而在 L_2 类型的扰动下，GGNA 方法相对于其他三种防御方法具有更高的对抗样本准确率。





(a)　L_∞ 类型的 PGD 攻击　　(b)　L_2 类型的 PGD 攻击

图 5　随着迭代次数的增加，四种防御方法的对抗样本准确率

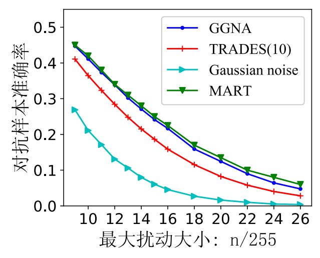
Fig. 5　As the number of iterations increases, the accuracy of three defense methods in adversarial examples

c) 最大扰动 ε 对对抗样本准确率的影响：图 6 显示了在 PGD 攻击下，随着最大扰动 ε 的增加，四种方法的对抗样本准确率比较。对于 L_∞ 类型的扰动，将最大扰动 ε_∞ 的范围设置为[8/255, 26/255]，而 L_2 类型的扰动，将最大扰动 ε_2 的范围设置为[3/255, 15/255]×32，将迭代次数 k 设置为 8，步长大小设置为 ε/k。从图 6 中可以看出，在 L_∞ 类型的扰动下，随着最大扰动的增加，GGNA 方法取得了和 MART 方法具有竞争力的对抗样本准确率，以及比另外两种方法更好的对抗样本准确率。在 L_2 类型的扰动下，随着最大扰动的增加，GGNA 方法相对于其他三种方法，都具有更高的对抗样本准确率。

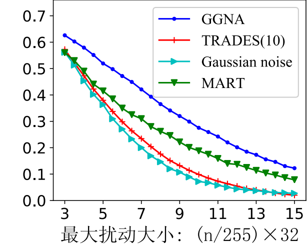
3.6.2 FGSM 和 Deepfool 攻击

本小节显示了在 Deepfool 和 FGSM 攻击下，三种防御方法的对抗样本准确率。对于 Deepfool 攻击，将最大迭代次数设置为 50 次，最大扰动设置为(4/255)×32。对于 $FGSM_2$ 攻击，将最大扰动设置为(4/255)×32，对于 $FGSM_\infty$ 攻击，将最大扰动设置为 8/255。由于 Deepfool 攻击中添加的扰动很少，所

以在测试 Deepfool 攻击时，对于 GGNA 和 Gaussian noise 方法，添加标准差为 0.1 的噪声。在表 4 中显示了四种防御方法在 FGSM 和 Deepfool 攻击下的准确率比较，可以看出 GGNA 方法在防御 L_2 类型的 FGSM 以及 Deepfool 攻击时，相对于另外三种防御方法具有较高的对抗样本准确率。



(a) L_∞ 类型的 PGD 攻击



(b) L_2 类型的 PGD 攻击

图 6　随着最大扰动的增加，四种防御方法的对抗样本准确率

Fig. 6　As the maximum disturbance increases, the accuracy of three defense methods in adversarial examples

表 4　FGSM 和 Deepfool 攻击下的对抗样本准确率

Tab. 4　Adversarial examples accuracy under FGSM and Deepfool attacks

方法	$FGSM_2$	$FGSM_\infty$	Deepfool
GGNA	0.5893	0.5051	0.8037
TRADES(10)	0.5457	0.5089	0.6856
Gaussian noise	0.4830	0.3519	0.7549
MART	0.5560	0.5112	0.6935

3.7　讨论

本文在 cifar-10 数据集上使用了多种攻击方法(PGD，FGSM，Deepfool)对提出的 GGNA 防御方法与多种防御方法(MART,TRADES,Gaussian noise)进行实验对比。由于 PGD 攻击方法产生的对抗样本更容易使分类系统分类错误，因此在该攻击下进行更加全面的测试，包括攻击的步长，迭代次数，最大扰动。实验结果表明，在 L_2 类型的扰动下，相对于其他三种防御方法，GGNA 方法都取得了更好的对抗样本准确率，而在 L_∞ 类型的扰动下，GGNA 方法也取得了和最新的 MART 防御方法相当的对抗样本准确率，相比 TRADES 方法和 Gaussian noise 方法更高的对抗样本准确率。总的来说，GGNA 方法有效地提高了图像分类模型分类对抗样本的准确率。

4　结束语

为了提高基于深度神经网络的图像分类模型防御对抗样本的能力，本文提出了梯度指导噪声添加(GGNA)的对抗训练方法。广泛的实验结果表明，相对于 TRADES，MART，Gaussian noise 方法，GGNA 方法在多种对抗样本的分类准确率上实现了相当或更好的性能，有效地提高了图像分类模型正确分类对抗样本的能力。

参考文献：

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]// Proc of the 26th Annual Conference on Neural Information Processing Systems, Nevada: MIT Press, 2012: 1097–1105.

[2] Hinton G, Deng Li, Yu Dong, *et al.* Deep neural networks for acoustic modeling in speech recognition [J]. IEEE Signal Processing Magazine, 2012, 29 (6): 1-29.

[3] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate [C]// Proc of the 3rd International Conference on Learning Representations, San Diego: IEEE Press, 2015: 46–61.

[4] Chen Chenyi, Seff Air, Kornhauser Alain, *et al.* DeepDriving: Learning affordance for direct perception in autonomous driving [C]// Proc of IEEE International Conference on Computer Vision, Santiago: IEEE Press, 2015: 2722–2730.

[5] Xu K, Ba J, Kiros R, *et al.* Show, attend and tell: Neural image caption generation with visual attention [C]// Proc of the 32nd International Conference on Machine Learning, Lille: ACM Press, 2015: 2048–2057.

[6] Szegedy C, Liu Wei, Jia Yangqing, *et al.* Going deeper with convolutions [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston: IEEE Press, 2015: 1–9.

[7] Szegedy C, Zaremba W, Sutskever I, *et al.* Intriguing properties of neural networks [C]// Proc of the 2nd International Conference on Learning Representations, Banff National Park: IEEE Press, 2014: 1-10.

[8] Xu Xiaojun, Chen Xinyun, Liu Chang, *et al.* Can you fool ai with adversarial examples on a visual turing test [J]. arXiv preprint arXiv: 1709. 08 693, 2017.

[9] Akhtar N, Mian A. Threat of Adversarial attacks on deep learning in computer Vision: a survey [J]. IEEE Access, 2018, 6 (4): 14410–14430.

[10] Chen Hongge, Zhang Huan, Chen Pinyu, *et al.* Show-and-fool: Crafting adversarial examples for neural image captioning [J]. arXiv preprint arXiv: 1712. 02051, 2017.

[11] Metzen J H, Kumar M C, Brox T, *et al.* Universal adversarial perturbations against semantic image segmentation [C]// Proc of the IEEE International Conference on Computer Vision, Venice: IEEE Press, 2017: 2774–2783.

[12] Zantedeschi V, Nicolae M I, Rawat A. Efficient defenses against adversarial atacks [C]// Proc of the 10th ACM Workshop on Artificial Intelligence and Security, colocated with CCS, 2017: 39–49.

[13] Liu Xuanqing, Cheng Minhao, Zhang Huan, *et al.* Towards robust neural networks via random self-ensemble [J]. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinfor-matics) , 2018, 11211 LNCS: 381–397.

[14] Papernot N, McDaniel P, Jha S, *et al.* The limitations of deep learning in adversarial settings [C]// IEEE European Symposium on Security and Privacy, Saarbrucken: IEEE Press, 2016: 372–387.

[15] Madry A, Makelov A, Schmidt L, *et al.* Towards deep learning models resistant to adversarial attacks [C]// Proc of the 6th International Conference on Learning Representations, Vancouver: IEEE Press, 2018: 1–28.

[16] Zhang Hongyang, Yu Yaodong, Jiao Jiantao, *et al.* Theoretically principled trade-off between robustness and accuracy [C]// Proc of the 36th International Conference on Machine Learning, California: ACM Press, 2019: 7472-7482.

[17] Kurakin A, Goodfellow I, Bengio S. Adversarial machine learning at scale [C]// Proc of the 5th International Conference on Learning Representations, Toulon: IEEE Press, 2017: 1–17.

[18] Moosavi-Dezfooli S M, Fawzi A, Frossard P. DeepFool: A simple and accurate method to fool deep neural networks [J]. Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas: IEEE Press, 2016: 2574–2582.

[19] Chen Pinyu, Zhang Huan, Sharma Yash, *et al.* ZOO: Zeroth order optimization based black-box atacks to deep neural networks without training substitute models [C]// Proc of the 10th ACM Workshop on Artificial Intelligence and Security, colocated with CCS, 2017: 15–26.

[20] Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples [C]// Proc of the 35th International Conference on Machine Learning, Stockholm: ACM Press, 2018: 436–448.

[21] Szegedy C, Zaremba W, Sutskever I, *et al.* Intriguing properties of neural

networks [C]// Proc of the 2nd International Conference on Learning Representations, Banff National Park: IEEE Press, 2014: 82–95.

[22] Pang Tianyu, Du Chao, Dong Yinpeng, *et al.* Towards robust detection of adversarial examples [C]// Proc of the 32nd Annual Conference on Neural Information Processing Systems, Montreal: MIT Press, 2018: 4579–4589.

[23] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images [R]. Citeseer, 2009.

[24] Tsipras D, Santurkar S, Engstrom L, *et al.* Robustness may be at odds with accuracy [C]// Proc of the 7th International Conference on Learning Representations, New Orleans: IEEE Press, 2019: 108-132.

[25] Wang Yisen, Zou Difan, Yi Jinfeng, *et al.* Improving Adversarial Robustness Requires Revisiting Misclassified Examples [C]// Proc of the 8th International Conference on Learning Representations, IEEE Press, 2020.

[26] 马玉琨, 毋立芳, 简萌, 等. 一种面向人脸活体检测的对抗样本生成算法 [J]. 软件学报, 2019, 30 (02): 469-480. (Ma Yukun, Wu Lifang, Jian Meng, *et al.* Algorithm to generate adversarial examples for face-spoofing detection. [J]. Journal of Software, 2019, 30 (2): 469−480.)

[27] 王文琦, 汪润, 王丽娜, 等. 面向中文文本倾向性分类的对抗样本生成方法 [J]. 软件学报, 2019, 30 (08): 2415-2427. (Wang Wenqi, Wang Run, Wang Lina, *et al.* Adversarial examples generation approach for tendency classification on chinese texts [J]. Journal of Software, 2019, 30 (8): 2415-2427.)